极客学院

Wiki > 移动开发 > iOS > The Swift Programming Language 中文版

(http://www.jikexueyuan.com)

关于 (http://wiki.jikexue yuan.com/project/swif t/)

欢迎使用 Swift (http://wi ki.jikexueyuan.com/proj ect/swift/chapter1/chapt er1.html)

Swift 教程 (http://wiki.ji kexueyuan.com/project/ swift/chapter2/chapter2. html)

基础部分 (http://wiki.ji kexueyuan.com/projec t/swift/chapter2/01_Th e_Basics.html)

基本运算符 (http://wiki. jikexueyuan.com/proje ct/swift/chapter2/02_B asic_Operators.html)

字符串和字符 (http://wi ki.jikexueyuan.com/pr oject/swift/chapter2/0 3_Strings_and_Charac ters.html)

集合类型 (http://wiki.ji kexueyuan.com/projec t/swift/chapter2/04_C ollection Types.html)

Via 由「极客学院 Wiki

(http://wiki.jikexueyuan.com)]

提供

集合类型 (Collection Types)

1.0 翻译: zqp (https://github.com/zqp) 校对: shinyzhu (https://github.com/shinyzhu), stanzhai (https://github.com/stanzhai), feiin (https://github.com/feiin)

2.0 翻译+校对: JackAlan (https://github.com/AlanMelody)

2.1 校对: shanks (http://codebuild.me)

2.2 校对: SketchK (https://github.com/SketchK) 2016-05-11

本页包含内容:

• 集合的可变性 (Mutability of Collections)

- 数组 (Arrays)
- 集合(Sets)
- 字典 (Dictionaries)

Swift 语言提供 Arrays 、 Sets 和 Dictionaries 三种基本的集合类型用来存储集合数据。数组(Arrays)是有序数据的集。集合(Sets)是无序无重复数据的集。字典(Dictionaries)是无序的键值对的集。

Swift 语言中的 Arrays 、 Sets 和 Dictionaries 中存储的数据值类型必须明确。这意味着我们不能把不正确的数据类型插入其中。同时这也说明我们完全可以对取回值的类型非常自信。

注意:

Swift 的 Arrays 、 Sets 和 Dictionaries 类型被实现为*泛型集合*。更多关于泛型类型和集合,参见 泛型 (./23_Generics.html)章节。

集合的可变性

如果创建一个 Arrays 、 Sets 或 Dictionaries 并且把它分配成一个变量,这个集合将会是*可变的*。这意味着我们可以在创建之后添加更多或移除已存在的数据项,或者改变集合中的数据项。如果我们把 Arrays 、 Sets 或 Dictionaries 分配成常量,那么它就是*不可变的*,它的大小和内容都不能被改变。

注意:

在我们不需要改变集合的时候创建不可变集合是很好的实践。如此 Swift 编译器可以优化我们创建的集合。

数组(Arrays)

数组使用有序列表存储同一类型的多个值。相同的值可以多次出现在一个数组的不同位置中。

注意: Swift 的 Array 类型被桥接到 Foundation 中的 NSArray 类。 更多关于在 Foundation 和 Cocoa 中使用 Array 的信息,参见 *Using Swift with Cocoa and Obejective-C*

(https://developer.apple.com/library/prerelease/ios/documentation/Swift/Conceptual/BuildingCocoaApps/index.htn
一书。

数组的简单语法

写 Swift 数组应该遵循像 Array<Element > 这样的形式,其中 Element 是这个数组中唯一允许存在的数据类型。我们也可以使用像 [Element] 这样的简单语法。尽管两种形式在功能上是一样的,但是推荐较短的那种,而且在本文中都会使用这种形式来使用数组。

创建一个空数组

我们可以使用构造语法来创建一个由特定数据类型构成的空数组:

```
var someInts = [Int]()
print("someInts is of type [Int] with \(someInts.count) items.")
// 打印 "someInts is of type [Int] with 0 items."
```

注意,通过构造函数的类型, someInts 的值类型被推断为 [Int] 。

(http://

(http://

(https

离线下载

PDF#

/downlo 相关资源 集合类型 (Collection Types) - The Swift Programming Language 中文版 - 极客学院Wiki

极客学院 jikexueyuan.com

(http://www.jikexueyuan.com)

关于 (http://wiki.jikexue yuan.com/project/swif t/)

欢迎使用 Swift (http://wi ki.jikexueyuan.com/proj ect/swift/chapter1/chapt er1.html)

Swift 教程 (http://wiki.ji kexueyuan.com/project/ swift/chapter2/chapter2. html)

基础部分 (http://wiki.ji kexueyuan.com/projec t/swift/chapter2/01_Th e_Basics.html)

基本运算符 (http://wiki. jikexueyuan.com/proje ct/swift/chapter2/02_B asic_Operators.html)

字符串和字符 (http://wi ki.jikexueyuan.com/pr oject/swift/chapter2/0 3_Strings_and_Charac ters.html)

集合类型 (http://wiki.ji kexueyuan.com/projec t/swift/chapter2/04_C ollection Types.html)

Via 由「极客学院 Wiki

(http://wiki.jikexueyuan.com)]

提供

或者,如果代码上下文中已经提供了类型信息,例如一个函数参数或者一个已经定义好类型的常量或者变量,我们可以使Wiki > 麻瓷或面裳句创建OS个全数面e Swistal Respenming Language 中文版

```
someInts.append(3)
// someInts 现在包含一个 Int 值
someInts = []
// someInts 现在是空数组,但是仍然是 [Int] 类型的。
```

创建一个带有默认值的数组

Swift 中的 Array 类型还提供一个可以创建特定大小并且所有数据都被默认的构造方法。我们可以把准备加入新数组的数据项数量(count)和适当类型的初始值(repeatedValue)传入数组构造函数:

```
var threeDoubles = [Double](count: 3, repeatedValue:0.0)
// threeDoubles 是一种 [Double] 数组,等价于 [0.0, 0.0, 0.0]
```

通过两个数组相加创建一个数组

我们可以使用加法操作符(+)来组合两种已存在的相同类型数组。新数组的数据类型会被从两个数组的数据类型中推断出来:

```
var anotherThreeDoubles = [Double](count: 3, repeatedValue: 2.5)
// anotherThreeDoubles 被推断为 [Double],等价于 [2.5, 2.5, 2.5]

var sixDoubles = threeDoubles + anotherThreeDoubles
// sixDoubles 被推断为 [Double],等价于 [0.0, 0.0, 0.0, 2.5, 2.5, 2.5]
```

用字面量构造数组

我们可以使用字面量来进行数组构造,这是一种用一个或者多个数值构造数组的简单方法。字面量是一系列由逗号分割并由方括号包含的数值:

[value 1, value 2, value 3] 。

下面这个例子创建了一个叫做 shoppingList 并且存储 String 的数组:

```
var shoppingList: [String] = ["Eggs", "Milk"]
// shoppingList 已经被构造并且拥有两个初始项。
```

shoppingList 变量被声明为"字符串值类型的数组",记作 [String] 。 因为这个数组被规定只有 String 一种数据结构,所以只有 String 类型可以在其中被存取。 在这里, shoppinglist 数组由两个 String 值("Eggs" 和 "Milk")构造,并且由字面量定义。

注意:

Shoppinglist 数组被声明为变量(var 关键字创建)而不是常量(let 创建)是因为以后可能会有更多的数据项被插入其中。

在这个例子中,字面量仅仅包含两个 String 值。匹配了该数组的变量声明(只能包含 String 的数组),所以这个字面 量的分配过程可以作为用两个初始项来构造 shoppinglist 的一种方式。

由于 Swift 的类型推断机制,当我们用字面量构造只拥有相同类型值数组的时候,我们不必把数组的类型定义清楚。 shoppinglist 的构造也可以这样写:

```
var shoppingList = ["Eggs", "Milk"]
```

因为所有字面量中的值都是相同的类型,Swift 可以推断出 [String] 是 shoppinglist 中变量的正确类型。

访问和修改数组

我们可以通过数组的方法和属性来访问和修改数组,或者使用下标语法。

可以使用数组的只读属性 count 来获取数组中的数据项数量·

```
print("The shopping list contains \((shoppingList.count) items.")
// 输出 "The shopping list contains 2 items."(这个数组有2个项)
```

使用布尔值属性 isEmpty 作为检查 count 属性的值是否为 0 的捷径:

(http://

(http://www.jikexueyuan.com)

关于 (http://wiki.jikexue yuan.com/project/swif t/)

欢迎使用 Swift (http://wi ki.jikexueyuan.com/proj ect/swift/chapter1/chapt er1.html)

Swift 教程 (http://wiki.ji kexueyuan.com/project/ swift/chapter2/chapter2. html)

基础部分 (http://wiki.ji kexueyuan.com/projec t/swift/chapter2/01_Th e_Basics.html)

基本运算符 (http://wiki. jikexueyuan.com/proje ct/swift/chapter2/02_B asic_Operators.html)

字符串和字符 (http://wi ki.jikexueyuan.com/pr oject/swift/chapter2/0 3_Strings_and_Charac ters.html)

集合类型 (http://wiki.ji kexueyuan.com/projec t/swift/chapter2/04_C ollection Types.html)

Via 由 [极客学院 Wiki

(http://wiki.jikexueyuan.com)]

提供

```
if shoppingList.isEmpty {

移动形ktとTinOShoppIngelSwiftsPEmpta.myming Language 中文版
} else {
    print("The shopping list is not empty.")
}

// 打印 "The shopping list is not empty." (shoppinglist 不是空的)
```

也可以使用 append(_:) 方法在数组后面添加新的数据项:

```
shoppingList.append("Flour")
// shoppingList 现在有3个数据项,有人在摊煎饼
```

除此之外,使用加法赋值运算符(+=)也可以直接在数组后面添加一个或多个拥有相同类型的数据项:

```
shoppingList += ["Baking Powder"]
// shoppingList 现在有四项了
shoppingList += ["Chocolate Spread", "Cheese", "Butter"]
// shoppingList 现在有七项了
```

可以直接使用下标语法来获取数组中的数据项,把我们需要的数据项的索引值放在直接放在数组名称的方括号中:

```
var firstItem = shoppingList[0]
// 第一项是 "Eggs"
```

注意:

第一项在数组中的索引值是 0 而不是 1 。 Swift 中的数组索引总是从零开始。

我们也可以用下标来改变某个已有索引值对应的数据值:

```
shoppingList[0] = "Six eggs"
// 其中的第一项现在是 "Six eggs" 而不是 "Eggs"
```

还可以利用下标来一次改变一系列数据值,即使新数据和原有数据的数量是不一样的。下面的例子把 "Chocolate Spread", "Cheese",和 "Butter"替换为 "Bananas"和 "Apples":

```
shoppingList[4...6] = ["Bananas", "Apples"]
// shoppingList 现在有6项
```

注意:

不可以用下标访问的形式去在数组尾部添加新项。

调用数组的 insert(_:atIndex:) 方法来在某个具体索引值之前添加数据项:

```
shoppingList.insert("Maple Syrup", atIndex: 0)
// shoppingList 现在有7项
// "Maple Syrup" 现在是这个列表中的第一项
```

这次 insert(_:atIndex:) 方法调用把值为 "Maple Syrup" 的新数据项插入列表的最开始位置,并且使用 0 作为索引值。

类似的我们可以使用 removeAtIndex(_:) 方法来移除数组中的某一项。这个方法把数组在特定索引值中存储的数据项移除并且返回这个被移除的数据项(我们不需要的时候就可以无视它):

```
let mapleSyrup = shoppingList.removeAtIndex(0)
// 索引值为0的数据项被移除
// shoppingList 现在只有6项,而且不包括 Maple Syrup
// mapleSyrup 常量的值等于被移除数据项的值 "Maple Syrup"
```

注意:

如果我们试着对索引越界的数据进行检索或者设置新值的操作,会引发一个运行期错误。我们可以使用索引值和数组的 count 属性进行比较来在使用某个索引之前先检验是否有效。除了当 count 等于 0 时(说明这是个空数组),最大索引值一直是 count - 1 ,因为数组都是零起索引。

数据项被移除后数组中的空出项会被自动填补,所以现在索引值为 0 的数据项的值再次等于 "Six eggs":

```
firstItem = shoppingList[0]
// firstItem 现在等于 "Six eggs"
```

(http://

(http://www.jikexueyuan.com)

关于 (http://wiki.jikexue yuan.com/project/swif

欢迎使用 Swift (http://wi ki.jikexueyuan.com/proj ect/swift/chapter1/chapt er1.html)

Swift 教程 (http://wiki.ji kexueyuan.com/project/ swift/chapter2/chapter2. html)

基础部分 (http://wiki.ji kexueyuan.com/projec t/swift/chapter2/01_Th e_Basics.html)

基本运算符 (http://wiki. jikexueyuan.com/proje ct/swift/chapter2/02_B asic_Operators.html)

字符串和字符 (http://wi ki.jikexueyuan.com/pr oject/swift/chapter2/0 3_Strings_and_Charac ters.html)

集合类型 (http://wiki.ji kexueyuan.com/projec t/swift/chapter2/04_C ollection Types.html)

Via 由「极客学院 Wiki

(http://wiki.jikexueyuan.com)]

提供

如果我们只想把数组中的最后一项移除,可以使用 removeLast() 方法而不是 removeAtIndex(_:) 方法来避免我们需要 Wiki > 表表现描解 couniQarte xindes with Proster () 方法来避免我们需要 wiki > 表表现描解 couniQarte xindes () 方法来避免我们需要

```
let apples = shoppingList.removeLast()
// 数组的最后一项被移除了
// shoppingList 现在只有5项,不包括 Apples
// apples 常量的值现在等于 "Apples" 字符串
```

数组的遍历

我们可以使用 for-in 循环来遍历所有数组中的数据项:

```
for item in shoppingList {
    print(item)
}
// Six eggs
// Milk
// Flour
// Baking Powder
// Bananas
```

如果我们同时需要每个数据项的值和索引值,可以使用 enumerate() 方法来进行数组遍历。 enumerate() 返回一个由每一个数据项索引值和数据值组成的元组。我们可以把这个元组分解成临时常量或者变量来进行遍历:

```
for (index, value) in shoppingList.enumerate() {
    print("Item \(String(index + 1)): \(value)")
}
// Item 1: Six eggs
// Item 2: Milk
// Item 3: Flour
// Item 4: Baking Powder
// Item 5: Bananas
```

更多关于 for-in 循环的介绍请参见for 循环 (05_Control_Flow.html#for_loops)。

集合(Sets)

*集合(Set)*用来存储相同类型并且没有确定顺序的值。当集合元素顺序不重要时或者希望确保每个元素只出现一次时可以使用集合而不是数组。

注意:

Swift的 Set 类型被桥接到 Foundation 中的 NSSet 类。

关于使用 Foundation 和 Cocoa 中 Set 的知识,请看 Using Swift with Cocoa and Objective-C

(https://developer.apple.com/library/prerelease/ios/documentation/Swift/Conceptual/BuildingCocoaApps/index.html (https://developer.apple.com

集合类型的哈希值

一个类型为了存储在集合中,该类型必须是可哈希化的--也就是说,该类型必须提供一个方法来计算它的哈希值。一个哈希值是 Int 类型的,相等的对象哈希值必须相同,比如 a==b ,因此必须 a .hashValue == b .hashValue 。

Swift 的所有基本类型(比如 String, Int, Double 和 Bool)默认都是可哈希化的,可以作为集合的值的类型或者字典的键的类型。没有关联值的枚举成员值(在枚举 (./08_Enumerations.html)有讲述)默认也是可哈希化的。

注意:

你可以使用你自定义的类型作为集合的值的类型或者是字典的键的类型,但你需要使你的自定义类型符合 Swift 标准库中的 Hashable 协议。符合 Hashable 协议的类型需要提供一个类型为 Int 的可读属性 hashValue 。由类型的 hashValue 属性返回的值不需要在同一程序的不同执行周期或者不同程序之间保持相同。

因为 Hashable 协议符合 Equatable 协议,所以符合该协议的类型也必须提供一个"是否相等"运算符(==)的实现。这个 Equatable 协议要求任何符合 == 实现的实例间都是一种相等的关系。也就是说,对于 a,b,c 三个值来说, == 的实现必须满足下面三种情况:

```
    a == a (自反性)
    a == b 意味着 b == a (对称性)
    a == b && b == c 意味着 a == c (传递性)
```

关于符合协议的更多信息,请看协议 (./22_Protocols.html)。

集合类型语法

(http://

集合类型 (Collection Types) - The Swift Programming Language 中文版 - 极客学院Wiki

极客学院 jikexueyuan.com Swift 中的 Set 类型被写为 Set<Element >,这里的 Element 表示 Set 中允许存储的类型,和数组不同的是,集合没有 Wiki > 新福斯格尔法。iOS > The Swift Programming Language 中文版

(http://

(http://www.jikexueyuan.com)

关于 (http://wiki.jikexue yuan.com/project/swif t/)

欢迎使用 Swift (http://wi ki.jikexueyuan.com/proj ect/swift/chapter1/chapt er1.html)

Swift 教程 (http://wiki.ji kexueyuan.com/project/ swift/chapter2/chapter2. html)

基础部分 (http://wiki.ji kexueyuan.com/projec t/swift/chapter2/01_Th e_Basics.html)

基本运算符 (http://wiki. jikexueyuan.com/proje ct/swift/chapter2/02_B asic_Operators.html)

字符串和字符 (http://wi ki.jikexueyuan.com/pr oject/swift/chapter2/0 3_Strings_and_Charac ters.html)

集合类型 (http://wiki.ji kexueyuan.com/projec t/swift/chapter2/04_C ollection Types.html)

Via 由「极客学院 Wiki

(http://wiki.jikexueyuan.com)]

提供

创建和构造一个空的集合

你可以通过构造器语法创建一个特定类型的空集合:

```
var letters = Set<Character>()
print("letters is of type Set<Character> with \(letters.count) items.")
// 打印 "letters is of type Set<Character> with 0 items."
```

注意:

通过构造器,这里的 letters 变量的类型被推断为 Set<Character>。

此外,如果上下文提供了类型信息,比如作为函数的参数或者已知类型的变量或常量,我们可以通过一个空的数组字面量创建一个空的 Set:

```
letters.insert("a")
// letters 现在含有1个 Character 类型的值
letters = []
// letters 现在是一个空的 Set, 但是它依然是 Set<Character> 类型
```

用数组字面量创建集合

你可以使用数组字面量来构造集合,并且可以使用简化形式写一个或者多个值作为集合元素。

下面的例子创建一个称之为 favoriteGenres 的集合来存储 String 类型的值:

```
var favoriteGenres: Set<String> = ["Rock", "Classical", "Hip hop"]
// favoriteGenres 被构造成含有三个初始值的集合
```

这个 favoriteGenres 变量被声明为"一个 String 值的集合",写为 Set<String>。由于这个特定的集合含有指定 String 类型的值,所以它只允许存储 String 类型值。这里的 favoriteGenres 变量有三个 String 类型的初始值 ("Rock", "Classical"和 "Hip hop"),并以数组字面量的方式出现。

注意:

favoriteGenres 被声明为一个变量(拥有 var 标示符)而不是一个常量(拥有 let 标示符),因为它里面的元素将会在下面的例子中被增加或者移除。

一个 Set 类型不能从数组字面量中被单独推断出来,因此 Set 类型必须显式声明。然而,由于 Swift 的类型推断功能,如果你想使用一个数组字面量构造一个 Set 并且该数组字面量中的所有元素类型相同,那么你无须写出 Set 的具体类型。 favoriteGenres 的构造形式可以采用简化的方式代替:

```
var favoriteGenres: Set = ["Rock", "Classical", "Hip hop"]
```

由于数组字面量中的所有元素类型相同,Swift 可以推断出 Set<String> 作为 favoriteGenres 变量的正确类型。

访问和修改一个集合

你可以通过 Set 的属性和方法来访问和修改一个 Set 。

为了找出一个 Set 中元素的数量,可以使用其只读属性 count:

```
print("I have \(favoriteGenres.count) favorite music genres.")
// 打印 "I have 3 favorite music genres."
```

使用布尔属性 is Empty 作为一个缩写形式去检查 count 属性是否为 0:

```
if favoriteGenres.isEmpty {
    print("As far as music goes, I'm not picky.")
} else {
    print("I have particular music preferences.")
}
// 打印 "I have particular music preferences."
```

你可以通过调用 Set 的 insert(_:) 方法来添加一个新元素:

Wiki >

(http://www.jikexueyuan.com)

关于 (http://wiki.jikexue yuan.com/project/swif t/)

欢迎使用 Swift (http://wi ki.jikexueyuan.com/proj ect/swift/chapter1/chapt er1.html)

Swift 教程 (http://wiki.ji kexueyuan.com/project/ swift/chapter2/chapter2. html)

基础部分 (http://wiki.ji kexueyuan.com/projec t/swift/chapter2/01_Th e_Basics.html)

基本运算符 (http://wiki. jikexueyuan.com/proje ct/swift/chapter2/02_B asic_Operators.html)

字符串和字符 (http://wi ki.jikexueyuan.com/pr oject/swift/chapter2/0 3_Strings_and_Charac ters.html)

集合类型 (http://wiki.ji kexueyuan.com/project/swift/chapter2/04_C ollection Types.html)

Via 由「极客学院 Wiki

(http://wiki.jikexueyuan.com)]

提供

```
favoriteGenres.insert("Jazz")
> 移动形态iteGiASs 观在图Swift累rogramming Language 中文版
```

你可以通过调用 Set 的 remove(_:) 方法去删除一个元素,如果该值是该 Set 的一个元素则删除该元素并且返回被删除的元素值,否则如果该 Set 不包含该值,则返回 nil。另外, Set 中的所有元素可以通过它的 removeAll() 方法删除。

```
if let removedGenre = favoriteGenres.remove("Rock") {
   print("\(removedGenre)? I'm over it.")
} else {
   print("I never much cared for that.")
}
// 打印 "Rock? I'm over it."
```

使用 contains (_:) 方法去检查 Set 中是否包含一个特定的值:

```
if favoriteGenres.contains("Funk") {
    print("I get up on the good foot.")
} else {
    print("It's too funky in here.")
}
// 打印 "It's too funky in here."
```

遍历一个集合

你可以在一个 for-in 循环中遍历一个 Set 中的所有值。

```
for genre in favoriteGenres {
    print("\(genre)")
}
// Classical
// Jazz
// Hip hop
```

更多关于 for-in 循环的信息,参见For 循环 (./05_Control_Flow.html#for_loops)。

Swift 的 Set 类型没有确定的顺序,为了按照特定顺序来遍历一个 Set 中的值可以使用 sort() 方法,它将根据提供的序列返回一个有序集合。

```
for genre in favoriteGenres.sort() {
    print("\(genre\)")
}
// prints "Classical"
// prints "Hip hop"
// prints "Jazz
```

集合操作

你可以高效地完成 Set 的一些基本操作,比如把两个集合组合到一起,判断两个集合共有元素,或者判断两个集合是否全包含,部分包含或者不相交。

基本集合操作

下面的插图描述了两个集合-a和b-以及通过阴影部分的区域显示集合各种操作的结果。

- 使用 intersect(_:) 方法根据两个集合中都包含的值创建的一个新的集合。
- 使用 $exclusive0r(_:)$ 方法根据在一个集合中但不在两个集合中的值创建一个新的集合。
- 使用 union(_:) 方法根据两个集合的值创建一个新的集合。
- 使用 subtract(_:) 方法根据不在该集合中的值创建一个新的集合。

```
let oddDigits: Set = [1, 3, 5, 7, 9]
let evenDigits: Set = [0, 2, 4, 6, 8]
let singleDigitPrimeNumbers: Set = [2, 3, 5, 7]

oddDigits.union(evenDigits).sort()
// [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
oddDigits.intersect(evenDigits).sort()
// []
oddDigits.subtract(singleDigitPrimeNumbers).sort()
// [1, 9]
oddDigits.exclusiveOr(singleDigitPrimeNumbers).sort()
// [1, 2, 9]
```

集合成员关系和相等

(http://

联,因为它们之间没有共同的元素。

(http://

(http://

关于 (http://wiki.jikexue yuan.com/project/swif t/)

(http://www.jikexueyuan.com)

欢迎使用 Swift (http://wi ki.jikexueyuan.com/proj ect/swift/chapter1/chapt er1.html)

Swift 教程 (http://wiki.ji kexueyuan.com/project/ swift/chapter2/chapter2. html)

基础部分 (http://wiki.ji kexueyuan.com/projec t/swift/chapter2/01_Th e_Basics.html)

基本运算符 (http://wiki. jikexueyuan.com/proje ct/swift/chapter2/02_B asic_Operators.html)

字符串和字符 (http://wi ki.jikexueyuan.com/pr oject/swift/chapter2/0 3_Strings_and_Charac ters.html)

集合类型 (http://wiki.ji kexueyuan.com/projec t/swift/chapter2/04_C ollection Types.html)

Via 由「极客学院 Wiki

(http://wiki.jikexueyuan.com)]

提供

- 使用"是否相等"运算符(==)来判断两个集合是否包含全部相同的值。
- 使用 isSubsetOf(_:) 方法来判断一个集合中的值是否也被包含在另外一个集合中。
- 使用 isSupersetOf(_:) 方法来判断一个集合中包含另一个集合中所有的值。
- 使用 isStrictSubsetOf(_:) 或者 isStrictSupersetOf(_:) 方法来判断一个集合是否是另外一个集合的子集合或者父集合并且两个集合并不相等。
- 使用 isDisjointWith(_:) 方法来判断两个集合是否不含有相同的值(是否没有交集)。

```
let houseAnimals: Set = [" ", " "]
let farmAnimals: Set = [" ", " ", " ", " ", " "]
let cityAnimals: Set = [" ", " "]
houseAnimals.isSubsetOf(farmAnimals)
// true
farmAnimals.isSupersetOf(houseAnimals)
// true
farmAnimals.isDisjointWith(cityAnimals)
// true
```

字典

字典是一种存储多个相同类型的值的容器。每个值(value)都关联唯一的键(key),键作为字典中的这个值数据的标识符。和数组中的数据项不同,字典中的数据项并没有具体顺序。我们在需要通过标识符(键)访问数据的时候使用字典,这种方法很大程度上和我们在现实世界中使用字典查字义的方法一样。

注意·

Swift 的 Dictionary 类型被桥接到 Foundation 的 NSDictionary 类。

更多关于在 Foundation 和 Cocoa 中使用 Dictionary 类型的信息,参见 *Using Swift with Cocoa and Objective-C (Swift 2.1)*

(https://developer.apple.com/library/prerelease/ios/documentation/Swift/Conceptual/BuildingCocoaApps/index.htn 一书。

字典类型快捷语法

Swift 的字典使用 Dictionary<Key, Value> 定义,其中 Key 是字典中键的数据类型, Value 是字典中对应于这些键所存储值的数据类型。

注意:

一个字典的 Key 类型必须遵循 Hashable 协议,就像 Set 的值类型。

我们也可以用 [Key: Value] 这样快捷的形式去创建一个字典类型。虽然这两种形式功能上相同,但是后者是首选,并且这本指导书涉及到字典类型时通篇采用后者。

创建一个空字典

我们可以像数组一样使用构造语法创建一个拥有确定类型的空字典:

```
var namesOfIntegers = [Int: String]()
// namesOfIntegers 是一个空的 [Int: String] 字典
```

这个例子创建了一个 [Int: String] 类型的空字典来储存整数的英语命名。它的键是 Int 型,值是 String 型。

如果上下文已经提供了类型信息,我们可以使用空字典字面量来创建一个空字典,记作[:](中括号中放一个冒号):

```
namesOfIntegers[16] = "sixteen"
// namesOfIntegers 现在包含一个键值对
namesOfIntegers = [:]
// namesOfIntegers 又成为了一个 [Int: String] 类型的空字典
```

用字典字面量创建字典

我们可以使用字典字面量来构造字典,这和我们刚才介绍过的数组字面量拥有相似语法。字典字面量是一种将一个或多个键值对写作 Dictionary 集合的快捷途径。

一个键值对是一个 key 和一个 value 的结合体。在字典字面量中,每一个键值对的键和值都由冒号分割。这些键值对构成一个列表,其中这些键值对由方括号包含、由逗号分割:

极客学院

[key 1: value 1, key 2: value 2, key 3: value 3] Wiki > 移动开发 > iOS > The Swift Programming Language 中文版

(http://

(http://

jikexueyuan.com

(http://www.jikexueyuan.com)

关于 (http://wiki.jikexue yuan.com/project/swif t/)

欢迎使用 Swift (http://wi ki.jikexueyuan.com/proj ect/swift/chapter1/chapt er1.html)

Swift 教程 (http://wiki.ji kexueyuan.com/project/ swift/chapter2/chapter2. html)

基础部分 (http://wiki.ji kexueyuan.com/projec t/swift/chapter2/01_Th e_Basics.html)

基本运算符 (http://wiki. jikexueyuan.com/proje ct/swift/chapter2/02_B asic_Operators.html)

字符串和字符 (http://wi ki.jikexueyuan.com/pr oject/swift/chapter2/0 3_Strings_and_Charac ters.html)

集合类型 (http://wiki.ji kexueyuan.com/project/swift/chapter2/04_C ollection Types.html)

Via 由「极客学院 Wiki

(http://wiki.jikexueyuan.com)]

提供

下面的例子创建了一个存储国际机场名称的字典。在这个字典中键是三个字母的国际航空运输相关代码,值是机场名称:

```
var airports: [String: String] = ["YYZ": "Toronto Pearson", "DUB": "Dublin"]
```

airports 字典被声明为一种 [String: String] 类型,这意味着这个字典的键和值都是 String 类型。

注意:

airports 字典被声明为变量(用 var 关键字)而不是常量(let 关键字)因为后来更多的机场信息会被添加到 这个示例字典中。

airports 字典使用字典字面量初始化,包含两个键值对。第一对的键是 YYZ ,值是 Toronto Pearson 。第二对的键是 DUB ,值是 Dublin 。

这个字典语句包含了两个 String: String 类型的键值对。它们对应 airports 变量声明的类型(一个只有 String 键和 String 值的字典)所以这个字典字面量的任务是构造拥有两个初始数据项的 airport 字典。

和数组一样,我们在用字典字面量构造字典时,如果它的键和值都有各自一致的类型,那么就不必写出字典的类型。 airports 字典也可以用这种简短方式定义:

```
var airports = ["YYZ": "Toronto Pearson", "DUB": "Dublin"]
```

因为这个语句中所有的键和值都各自拥有相同的数据类型,Swift 可以推断出 Dictionary<String, String> 是 airports 字典的正确类型。

访问和修改字典

我们可以通过字典的方法和属性来访问和修改字典,或者通过使用下标语法。

和数组一样,我们可以通过字典的只读属性 count 来获取某个字典的数据项数量:

```
print("The dictionary of airports contains \(airports.count) items.")
// 打印 "The dictionary of airports contains 2 items."(这个字典有两个数据项)
```

使用布尔属性 isEmpty 来快捷地检查字典的 count 属性是否等于0:

```
if airports.isEmpty {
    print("The airports dictionary is empty.")
} else {
    print("The airports dictionary is not empty.")
}
// 打印 "The airports dictionary is not empty."
```

我们也可以在字典中使用下标语法来添加新的数据项。可以使用一个恰当类型的键作为下标索引,并且分配恰当类型的新值:

```
airports["LHR"] = "London"
// airports 字典现在有三个数据项
```

我们也可以使用下标语法来改变特定键对应的值:

```
airports["LHR"] = "London Heathrow"
// "LHR"对应的值 被改为 "London Heathrow
```

作为另一种下标方法,字典的 updateValue(_:forKey:) 方法可以设置或者更新特定键对应的值。就像上面所示的下标示例, updateValue(_:forKey:) 方法在这个键不存在对应值的时候会设置新值或者在存在时更新已存在的值。和上面的下标方法不同的, updateValue(_:forKey:) 这个方法返回更新值之前的原值。这样使得我们可以检查更新是否成功。

updateValue(_:forKey:) 方法会返回对应值的类型的可选值。举例来说:对于存储 String 值的字典,这个函数会返回一个 String? 或者"可选 String"类型的值。

如果有值存在于更新前,则这个可选值包含了旧值,否则它将会是 nil 。

```
if let oldValue = airports.updateValue("Dublin Airport", forKey: "DUB") {
   print("The old value for DUB was \((oldValue).")
}
// 输出 "The old value for DUB was Dublin."
```

我们也可以使用下标语法来在字典中检索特定键对应的值。因为有可能请求的键没有对应的值存在,字典的下标访问会返 Wiki > 直發動面類类型的分洗值。如果这种時期也多數學與的例如由他如果是透面版个包含这个存在值的可选值,否则将返回

(http://

(http://

关于 (http://wiki.jikexue yuan.com/project/swif

(http://www.jikexueyuan.com)

欢迎使用 Swift (http://wi ki.iikexuevuan.com/proi ect/swift/chapter1/chapt er1.html)

Swift 教程 (http://wiki.ji kexueyuan.com/project/ swift/chapter2/chapter2. html)

基础部分 (http://wiki.ji kexueyuan.com/projec t/swift/chapter2/01_Th e Basics.html)

基本运算符 (http://wiki. jikexueyuan.com/proje ct/swift/chapter2/02_B asic Operators.html)

字符串和字符 (http://wi ki.jikexueyuan.com/pr oject/swift/chapter2/0 3_Strings_and_Charac ters.html)

集合类型 (http://wiki.ji kexueyuan.com/projec t/swift/chapter2/04 C ollection Types.html)

Via 由「极客学院 Wiki

(http://wiki.jikexueyuan.com)]

提供

```
if let airportName = airports["DUB"] {
   print("The name of the airport is \((airportName).")
} else {
   print("That airport is not in the airports dictionary.")
// 打印 "The name of the airport is Dublin Airport."
```

我们还可以使用下标语法来通过给某个键的对应值赋值为 nil 来从字典里移除一个键值对:

```
airports["APL"] = "Apple Internation"
// "Apple Internation" 不是真的 APL 机场, 删除它
airports["APL"] = nil
// APL 现在被移除了
```

此外, removeValueForKey(_:) 方法也可以用来在字典中移除键值对。这个方法在键值对存在的情况下会移除该键值对 并且返回被移除的值或者在没有值的情况下返回 nil:

```
if let removedValue = airports.removeValueForKey("DUB") {
   print("The removed airport's name is \((removedValue).")
} else {
   print("The airports dictionary does not contain a value for DUB.")
// prints "The removed airport's name is Dublin Airport."
```

字典遍历

我们可以使用 for-in 循环来遍历某个字典中的键值对。每一个字典中的数据项都以 (key, value) 元组形式返回,并且 我们可以使用临时常量或者变量来分解这些元组:

```
for (airportCode, airportName) in airports {
   print("\(airportCode): \(airportName)")
// YYZ: Toronto Pearson
// LHR: London Heathrow
```

更多关于 for-in 循环的信息,参见For 循环 (./05_Control_Flow.html#for_loops)。

通过访问 keys 或者 values 属性,我们也可以遍历字典的键或者值:

```
for airportCode in airports.keys {
   print("Airport code: \(airportCode)")
// Airport code: YYZ
// Airport code: LHR
for airportName in airports.values {
    print("Airport name: \(airportName)")
// Airport name: Toronto Pearson
// Airport name: London Heathrow
```

如果我们只是需要使用某个字典的键集合或者值集合来作为某个接受 Array 实例的 API 的参数,可以直接使用 keys 或 者 values 属性构造一个新数组:

```
let airportCodes = [String](airports.keys)
// airportCodes 是 ["YYZ", "LHR"]
let airportNames = [String](airports.values)
// airportNames 是 ["Toronto Pearson", "London Heathrow"]
```

Swift 的字典类型是无序集合类型。为了以特定的顺序遍历字典的键或值,可以对字典的 keys 或 values 属性使 用 sort() 方法。

上一篇:字符串和字符 (/project/swift/chapter2/03_Strings_and_Characters.html) 下一篇: 控制流 (/project/swift/chapter2/05_Control_Flow.html)

```
被顶起来的评论
```

极客学院

Wiki >

(http://www.jikexueyuan.com)

关于 (http://wiki.jikexue yuan.com/project/swif

欢迎使用 Swift (http://wi ki.iikexuevuan.com/proi ect/swift/chapter1/chapt er1.html)

Swift 教程 (http://wiki.ji kexueyuan.com/project/ swift/chapter2/chapter2. html)

基础部分 (http://wiki.ji kexueyuan.com/projec t/swift/chapter2/01_Th e_Basics.html)

基本运算符 (http://wiki. jikexueyuan.com/proje ct/swift/chapter2/02_B asic Operators.html)

字符串和字符 (http://wi ki.jikexueyuan.com/pr oject/swift/chapter2/0 3_Strings_and_Charac ters.html)

集合类型 (http://wiki.ji kexueyuan.com/projec t/swift/chapter2/04_C ollection Types.html)

Via 由「极客学院 Wiki

(http://wiki.jikexueyuan.com)]

提供

移动开发//shippingListThe Swift, Programming Language 中文版

这句注释写错了应该是

(http://4jplus.com/)2015年8月7日

// shoppingList 现在只有5项,不包括 apples

2015年12月7日 回复 顶(1)

Vincent 4J (http://4jplus.com/) 收到,稍后在底部加个「下一章」按钮

32条评论 1条新浪微博

最新 最早 最执

(http://

(http://

Realank刘 (http://weibo.com/realank) 我就想知道,从网络获取的数据model,是什么样的字典类型。。。。

(http://weibo.com/realank)1月14日 回复 顶

Realank刘 (http://weibo.com/realank)

顶(1)

回复 孟海盗: 好的我来改

回复

http://weibo.com/realank)1月14日 回复 转发



iack

如何定义一个有大小的空数组

1月13日 回复 顶



孟海盗

// shoppingList 现在只有5项,不包括 cheese

这句注释写错了应该是

// shoppingList 现在只有5项,不包括 apples

2015年12月7日 回复 顶(1)



从今以后 (http://t.qq.com/congjinyih6382)

回复 热心的游客: 这个应该是 Swift 2.0 的语法, Xcode 7 支持。之前是用 (http://t.qq.com/congjinyih6382) enumerate(shoppingList) 函数,现在变成属性了。

2015年11月16日 回复

从今以后 (http://t.qq.com/congjinyih6382)

回复 whoever: 我感觉 "完全不含有" 和 "不含有" 是一个意思啊,只要含有一个就不能

(http://t.qq.com/congjinyih6382)^{算是"不含有"了}

2015年11月16日 回复 顶 转发



试了一下 isDisjointWith,应该是判断两个集合是否 完 全 不含有相同的值,在文章里的表述是 判断两个集合是否 不含有相同的值,不是很准确。

2015年11月9日 回复 转发



肖泽 (http://weibo.com/obamamamama)

回复 白宇努力考试读书养看看: 大神推荐本书来

weibo.com/obamamamama)2015年11月5日



王忠星

'关于使用Fundation和Cocoa中Set的知识', 应该是Foundation吧?

2015年10月30日 回复 顶 转发



沉沦2013

回复潘:原来是有7项的,这样相当于把shoppingList[4...6]的三项换成了两项,改变后只有6项了

2015年10月25日 回复



回复 爱看云: EPUB 版本的 排版和目录都有问题

2015年10月19日 回复 顶 转发



白宇努力考试读书养看看 (http://weibo.com/2293862303)

回复 大明小书童: 我当时用6写的,现在的7取消了。从函数变成method了

(http://weibo.com/2293862303)2015年9月26日



大明小书童 (http://t.qq.com/kjkqjll)

回复 白宇努力考试读书养看看: 你是哪个版本的xocode。确定有这个函数吗?

http://t.qq.com/kjkqjll)2015年9月23日 回复 转发



Java转Swift, 感觉Swift语法简洁, 功能强大, 一点都不拖泥带水, 越来越喜欢了。

2015年9月20日 回复 转发

Wiki >

(http://www.jikexueyuan.com)

关于 (http://wiki.jikexue yuan.com/project/swif

欢迎使用 Swift (http://wi ki.iikexuevuan.com/proi ect/swift/chapter1/chapt er1.html)

Swift 教程 (http://wiki.ji kexueyuan.com/project/ swift/chapter2/chapter2. html)

基础部分 (http://wiki.ji kexueyuan.com/projec t/swift/chapter2/01_Th e_Basics.html)

基本运算符 (http://wiki. jikexueyuan.com/proje ct/swift/chapter2/02_B asic Operators.html)

字符串和字符 (http://wi ki.jikexueyuan.com/pr oject/swift/chapter2/0 3_Strings_and_Charac ters.html)

集合类型 (http://wiki.ji kexueyuan.com/projec t/swift/chapter2/04_C ollection Types.html)

Via 由 [极客学院 Wiki

(http://wiki.jikexueyuan.com)]

提供

移动开发^{爱看云}OS > The Swift Programming Language 中文版

离线下载 EPUB版本下载了里面显示到 字符串和字符 章节 后面都不能显示了

2015年9月16日 回复 顶 转发

> 如果我瘦了请记得我曾经胖过 (http://weibo.com/1671342070) 小节:集合的可变性

(http://weibo.com/1671342070) 如果我们把Arrays、Sets或Dictionaries分配成常量,那么它就是不可变的,它的大小不 能被改变。

转发

这部分有翻译错误,官方文档是说size and contents can not be changed。 我试了半天,内容和长度都是无法改变的。

2015年9月14日 回复

king (http://t.qq.com/jj5671202)

拜谢各翻译人员

(http://t.qq.com/jj5671202)_{2015年9月8日} 回复

denawei1999

let oddDigits: Set = [1,3,5,7,9]let eventDigits: Set = [0,2,4,6,8]

let singleDigitPrimeNumber: Set = [2,3,5,7]

print("排序后的数组\(sorted(oddDigits.union(eventDigits)))\n")

print((sorted(oddDigits.intersect(eventDigits))))

print(sorted(oddDigits.subtract(singleDigitPrimeNumber))) print(sorted(oddDigits.exclusiveOr(singleDigitPrimeNumber)))

2015年8月26日 回复 顶 转发

王学森 (http://weibo.com/imwangxuesen)

回复 白宇努力考试读书养看看: 我下载了xcode7 beta5 .sort()方法是可以用的,所以

(http://weibo.com/imwangxuesen)如果你用的是xcode6应该是swift只是支持到1.0 没有到2.0 所以当前介绍的一些语法 **,**时不能编译的

2015年8月21日 回复 顶 转发

白宇努力考试读书养看看 (http://weibo.com/2293862303)

回复 王学森: set中用sorted (_:)函数你试试

(http://weibo.com/2293862303)2015年8月21日 回复 顶

社交帐号登录: 微信 微博 QQ 人人 更多»

说点什么吧...

「极客学院 Wiki - wiki.jikexueyuan.com」正在使用多说 (http://duoshuo.com)

(http://

(http://

1 2

发布